General (28)
(8) Sketch out a graph of your game (include exits). Decide which room lets you "win".
(12) create a game class and implement your Places and Exits.
(8) Revise the input system so the player can make multi actions without being taken to a new place.

Implement Your Own Game (=28+?)
(8) Must have at least 8 Places.
(8) Must have at least 8 functional Exits.
(4) Must have a terminal place (a way to win the game).
(4) Have some one-way paths (the staircase collapses behind you)
(4) Have some rooms that are locked-LockedExit

Create a Player class (32)
(4) The player class has the enterRoom() function.
(4) The player class has the pickUpItem() function.
(4) The player class has the useItem() function.
(4) The player class has the attribute HP, which decreases as the player object calls a function, when it's 0, the player loses the game.
(4) The player has a way to check what items he/she's bringing.
(4) Make a stuff command that prints out the users items or "You have nothing."
(4) The player can get rid of items.
(4) The user has a checkTime( ) function with which he/she can check the current time.

Create an Item class (28)
(4) Items have names and descriptions.
(4) Place must have a method that returns a list of items in a location (maybe just String?)
(4) Items are part of the description until they are taken from a Place.
(4) The item discarded by the player will be added into the place description.
(8) The user can combine two items to make a new item.
(4) There's a way to destroy an item.

Create a phone class (24)
(4) At some point the player can find a phone, he'll find that the phone doesn't have battery
(4) The player can find the buttery and put them together.
(8) The phone is locked. But the player can unlock it using a password.
(4) The phone will receive messages after it's unlocked.
(4) The message comes regardless of the player's movements.

Key and LockedExit (20)
(8) Create a Key class which extends the Item class. The player can open doors with keys.
(12) A LockedExit that can only be chosen with the appropriate Key.

Implement SecretExit (=24)
(4) Create a class SecretExit that extends Exit.

? What both classes? Anyway I don't need this.

~~(4) Put a method called "isSecret" on Exit that works for both classes.~~
(4) SecretExit should have a boolean hidden, that starts off as true.
(4) Make it so SecretExits are not printed to the user (in InteractiveFiction) when hidden.
(4) When a user types "search", if there is a SecretExit in the room they are currently in, it should be made visible to them.
~~(4) Put a SecretExit from the basement to the secretRoom in SpookyMansion, so that the dumbwaiter is not the only solution to the game.~~

Create a time system in your game. (=20)
(8) Create a class GameTime, that has an int as state, ranging from 0 to 23 (inclusive).
(4) Extend the game's concept of a player with a current time.
(4) Increment the hour regardless of the user's movement.
(4) Tell the player how many hours they spent in your game upon GameOver.

Implement Different Place Descriptions for Day and Night (=24)
(4) Make an action rest that advances the game 2 hours so that you can test your descriptions.
(4) Implement a boolean method called isNightTime()
(4) Add GameTime as a parameter to the getDescription method on Place.
(8) Create some places with descriptions based whether GameTime isNightTime() -- have a new nightDescription field be null by default.
(4) Have some items only function at night. ->torch, logic in player.useItem()

(12) Challenge: I'm not breaking this down, shares some work with SecretExit.
Implement a new game with some other tool or switch or idea (=?)

New:

(4)Food class
when combine 2 food item together, there's chance to generate poisonous food which decreases player.HP
(4)Some action of the player will lead to the end of the game->Using torch during the night in the wardrobeRoom will trigger the happy end.